

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-298593

(43)Date of publication of application : 24.10.2000

(51)Int.Cl.

G06F 9/46
G06F 11/28

(21)Application number : 11-106033

(71)Applicant : NEC CORP

(22)Date of filing : 14.04.1999

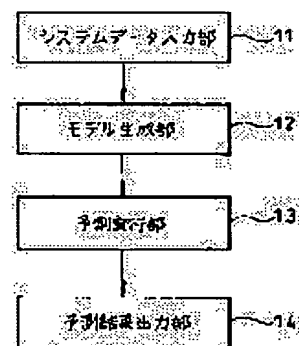
(72)Inventor : KURASUGI TOSHIYASU

(54) SYSTEM AND METHOD FOR PREDICTING PERFORMANCE OF MULTITASK SYSTEM AND RECORDING MEDIUM STORED WITH PROGRAM FOR THE METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To accurately predict a performance index for the parallel degree of parallel computers in a multitask environment.

SOLUTION: When the specification of a parallel degree or the like of parallel computers to be a platform is inputted to an input part 11 in the performance prediction system, a model generation part 12 generates a model on the basis of the specification. A prediction execution part 13 calculates performance index prediction values such as the degree of improvement and more detailed performance indexes such as throughput, a response and a resource using rate from the generated model. A prediction result output part 14 visually expresses the values outputted from the execution part 13.



LEGAL STATUS

[Date of request for examination] 24.03.2000

[Date of sending the examiner's decision of rejection] 10.08.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3663968

[Date of registration] 08.04.2005

[Number of appeal against examiner's decision of rejection] 2004-18617

[Date of requesting appeal against examiner's decision of rejection] 09.09.2004

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-298593

(P2000-298593A)

(43) 公開日 平成12年10月24日 (2000. 10. 24)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 9/46	3 6 0	G 0 6 F 9/46	3 6 0 C 5 B 0 4 2
11/28	3 4 0	11/28	3 4 0 C 5 B 0 9 8

審査請求 有 請求項の数13 O L (全 9 頁)

(21) 出願番号 特願平11-106033

(22) 出願日 平成11年4月14日 (1999. 4. 14)

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 藤杉 俊康

東京都港区芝五丁目7番1号 日本電気株式会社内

(74) 代理人 100088812

弁理士 ▲柳▼川 信

Fターム (参考) 5B042 GA11 GA23 HH07 HH20 MC28
NN04 NN08

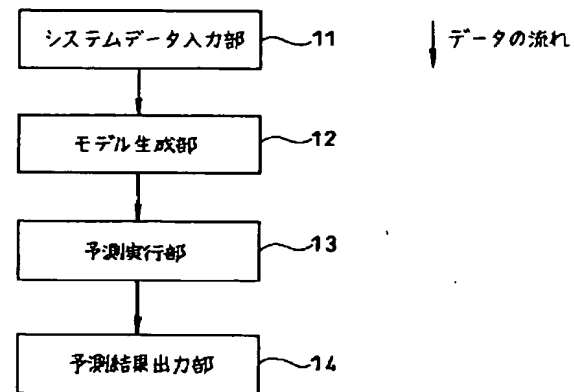
5B098 AA10 GA04 GB13 GC10 GD15

(54) 【発明の名称】 マルチタスクシステムの性能予測システム及び予測方法並びにその方法プログラムを記録した記録媒体

(57) 【要約】

【課題】 マルチタスク環境における並列計算機の並列度に対する性能指標を正しく予測する。

【解決手段】 プラットホームとなる並列計算機の並列度等のスペックが入力部11に入力されると、モデル生成部12はこのスペックよりモデルを生成する。予測実行部13は生成されたモデルから向上度やより細かい性能指標であるスループット、レスポンス、リソース使用率などの性能指標予測値を算出する。予測結果出力部14は予測実行部13から出力された値を可視的に表現する。



【特許請求の範囲】

【請求項 1】 並列計算機をプラットフォームとするマルチタスクシステムの性能予測システムであって、マルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化するモデル化手段と、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する性能指標予測手段とを含むことを特徴とするマルチタスクシステムの性能予測システム。

【請求項 2】 前記モデル化手段は前記並列計算機のスベックが入力されるデータ入力部と、このデータ入力部に入力されたデータを用いてモデルを生成するモデル生成部とを含むことを特徴とする請求項 1 記載の性能予測システム。

【請求項 3】 前記性能指標予測手段は前記モデル化手段にてモデル化されたリソースの競合を解析し性能指標を予測する予測実行部と、この予測実行部での予測結果を出力する予測結果出力部とを含むことを特徴とする請求項 1 又は 2 記載の性能予測システム。

【請求項 4】 前記モデル化手段はプラットフォームのスベックをパラメータとしてモデル化を行い、前記性能指標予測手段はそのパラメータに対する性能指標を予測することを特徴とする請求項 1 乃至 3 いずれかに記載の性能予測システム。

【請求項 5】 前記モデル化手段はセマフォやクリティカルセクションをはじめとした実行されるプログラムの構造をパラメータとしてモデル化を行い、前記性能指標予測手段はそのパラメータに対する性能指標を予測することを特徴とする請求項 1 乃至 3 いずれかに記載の性能予測システム。

【請求項 6】 前記モデル化手段は同時に実行されるプログラム又はタスクの数、スレッドの数などのシステム負荷をパラメータとしてモデル化を行い、前記性能指標予測手段はそのパラメータに対する性能指標を予測することを特徴とする請求項 1 乃至 3 いずれかに記載の性能予測システム。

【請求項 7】 前記性能指標予測手段は前記待ち行列網モデルの状態の変化を各タスク又はスレッドの処理速度から追うことにより前記待ち行列網モデルのスループットを性能予測値として算出することを特徴とする請求項 1 乃至 6 いずれかに記載の性能予測システム。

【請求項 8】 並列計算機をプラットフォームとするマルチタスクシステムの性能予測方法であって、マルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化する第 1 ステップと、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する第 2 ステップとを含むことを特徴とするマルチタスクシステムの性能予測方法。

【請求項 9】 前記第 1 ステップは前記並列計算機のス

ベックをデータ入力部に入力する第 1 ステップと、このデータ入力部に入力されたデータを用いてモデルを生成する第 2 ステップとを含むことを特徴とする請求項 8 記載の性能予測方法。

【請求項 10】 前記第 2 ステップは前記第 1 ステップにてモデル化されたリソースの競合を解析し性能指標を予測する第 2 ステップと、この第 2 ステップでの予測結果を出力する第 2 ステップとを含むことを特徴とする請求項 8 又は 9 記載の性能予測方法。

【請求項 11】 並列計算機をプラットフォームとするマルチタスクシステムの性能予測方法プログラムを記録した記録媒体であって、マルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化する第 1 ステップと、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する第 2 ステップとを含むプログラムを記録した記録媒体。

【請求項 12】 前記第 1 ステップは前記並列計算機のスベックをデータ入力部に入力する第 1 ステップと、このデータ入力部に入力されたデータを用いてモデルを生成する第 2 ステップとを含むことを特徴とする請求項 11 記載の記録媒体。

【請求項 13】 前記第 2 ステップは前記第 1 ステップにてモデル化されたリソースの競合を解析し性能指標を予測する第 2 ステップと、この第 2 ステップでの予測結果を出力する第 2 ステップとを含むことを特徴とする請求項 11 又は 12 記載の記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はマルチタスクシステムの性能予測システム及び予測方法並びにその方法プログラムを記録した記録媒体に関し、特に並列計算機をプラットフォームとするマルチタスクシステムの性能予測システム及び予測方法並びにその方法プログラムを記録した記録媒体に関する。

【0002】

【従来の技術】並列計算機における向上度（並列度（同時に実行状態になるタスクやスレッドの最大数）に対する処理性能の向上を示す指標）の予測手法が 1967 年に発行された AFIPS Conference Proceedings の第 483～485 頁に掲載された "Validity of the single-processor approach to achieving largescale computing" と題する G. Amdahl による論文に掲載されている。この従来の方法においては、シングルタスク環境が想定されており、その仮定のもとに、並列度に対する処理性能の向上度の予測手法が示されている。

【0003】又、1988 年に発行された CACM Vol. 31 (5) の第 532～533 頁に掲載された"

Reevaluating Amdahl's Law"と題するJ. Gustafsonによる論文、1996年に発行された並列処理シンポジウムの第227～234頁に掲載された「高並列計算機の性能評価のための挙動予測モデルの構成」と題する古市らによる論文にも同様にシングルタスク環境の仮定をおいた向上度の予測手法が示されている。

【0004】又、マルチタスク環境を想定した場合の予測手法が、1996年に発行されたCommun. ACM 39 (12) のArticle 231に掲載された" Including Queueing Effects in Amdahl's Law"と題するR. Nelsonによる論文に掲載されている。この論文ではマルチタスク環境によるタスク（又はスレッド）間のプロセッサ競合を考慮したパラメータを予測式に取り入れている。

【0005】ところで、マルチプロセッサを搭載したSMP (symmetrical multiprocessor; 対称型マルチプロセッサ) マシンが比較的安価になるに伴い、コンピュータシステムのプラットフォーム (platform) に採用されるなど、従来は大規模な科学計算などに用いられていた並列計算機が身近なものになりつつある。

【0006】システム開発において、並列計算機をプラットフォームとして採用する場合に問題となることの1つが、並列計算機において並列度をどの程度に設定するかである。並列度は計算機におけるプロセッサの数と一致することが多く、一般には並列度をあげると性能は向上するが、このことはプロセッサ数の増加に繋がるため、プラットフォームに対するコストは増加してしまう。

【0007】コストパフォーマンスの高いシステム開発を行うためには、適度な並列度の設定を行うこと、さらには（要求された処理性能を満たす範囲内で）できるだけ並列度を低くするようなシステム設計が可能であることが望ましい。そのためには、システム構成を与えた場合に並列度に対する処理性能の指標（具体的には速度向上率、効率、スループット (throughput; 単位時間当たりの処理能力) やレスポンスタイム、資源使用率などを指す。）を予測する必要がある。このような指標を予測する場合、従来は主にシングルタスク環境での利用を想定して行われていた。シングルタスク環境下での速度向上率や効率（これらの指標の定義は上記古市らの論文の第228項参照のこと）の予測手法が上記のようにG. Amdahlによる論文をはじめとした論文で示されている。また、マルチタスク環境を想定し、向上度を予測する手法も上記R. Nelsonの論文で示されている。

【0008】図8は並列計算機をプラットフォームとするマルチタスク環境でのコンピュータシステムの一例を示す構成図である。図8を参照して、並列計算機101は

複数のプロセッサ、この例では3つのプロセッサ102～104を有している。一方、実行される複数のタスクとして、この例では4つのタスク105～108が存在する。このコンピュータシステムでは、複数のプロセッサ102～104を搭載したコンピュータ（並列計算機）101上で、同時に複数のタスク105～108の実行が行われるのである。

【0009】又、この種の技術の一例が特開平9-237203号公報、特開昭62-182864号公報、特開昭59-174957号公報及び特開平10-069469号公報に開示されている。

【0010】

【発明が解決しようとする課題】第1の問題点は、マルチタスク環境又はマルチスレッド環境における並列計算機の並列度に対する性能の向上度や性能指標を正しく予測できないということである。その理由は、シングルタスク環境を想定して行った予測では、複数のタスクによるプロセッサの競合などのリソース競合を考慮していないので、誤った予測しか行えないためである。また、R. Nelsonの論文において示されている手法では、予測式の根拠となる理論が示されていないためにその精度に疑問があるからである。

【0011】第2の問題点は、コストにふさわしい並列計算機のスペック、より具体的には並列度やプロセッサの処理速度といったものの決定を勘や経験にたよるを得ないということである。その理由は、第1の問題点に示されるように性能指標を適切に予測することができないためである。

【0012】第3の問題点は、並列計算機上で実行されるプログラムの構造にボトルネック (bottle neck; 阻害要因) があってもその箇所の特定ができないことである。その理由は、プログラムではセマフォ (semaphore) やクリティカルセクション (critical section) などを用いた制御が行われており、このことがボトルネックとなることが多々ある。しかしながら、従来の方法では、マルチタスク環境でのこれらプログラムの構造を適切に考慮していないということが理由である。なお、「セマフォ」とは複数のタスクが同時に動く場合にタスク間の同期を取るときに使う信号のことをいい、「クリティカルセクション」とは複数のタスクが同時実行される環境において、あるタスクがクリティカルな（重要な）資源を使用している間は他のタスクにその資源が解放されるまでその資源の使用を待たせること、即ちその待たせる期間のことをいう。

【0013】第4の問題点は要求性能を満足するため、同時に実行するプログラムの構成やタスク数、スレッド数といったシステム負荷の設定を勘や経験のみで行わなければならないということである。その理由は、従来の方法にはマルチタスク環境又はマルチスレッド環境

におけるシステムの性能指標を適切に予測するためのモデルおよびその解析手法が示されていないためである。

【0014】第5の問題点は、システムにおけるスループット、レスポンスタイム（response time；応答時間）、リソース（resource；資源）の使用率といったより細かい性能指標の予測が不可能であることである。その理由は、従来の方法では解析を容易にするために、算出する指標を速度向上率や効率などの特定されたものに限定してしまっているからである。

【0015】そこで本発明の目的は、上記課題を解決することが可能なマルチタスクシステムの性能予測システム及び予測方法並びにその方法プログラムを記録した記録媒体を提供することにある。

【0016】

【課題を解決するための手段】前記課題を解決するために本発明による第1の発明は、並列計算機をプラットフォームとするマルチタスクシステムの性能予測システムであって、そのシステムはマルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化するモデル化手段と、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する性能指標予測手段とを含むことを特徴とする。

【0017】又、本発明による第2の発明は、並列計算機をプラットフォームとするマルチタスクシステムの性能予測方法であって、その方法はマルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化する第1ステップと、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する第2ステップとを含むことを特徴とする。

【0018】又、本発明による第3の発明は、並列計算機をプラットフォームとするマルチタスクシステムの性能予測方法プログラムを記録した記録媒体であって、その記録媒体にはマルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化する第1ステップと、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する第2ステップとを含むプログラムが記録されていることを特徴とする。

【0019】第1乃至第3の発明によれば、まず第1にマルチタスク環境又はマルチスレッド環境における並列計算機の並列度に対する性能の向上度や性能指標を正しく予測することができる。次に第2にかけられるコストにふさわしい並列度、より具体的にはプロセッサの数の決定を適切に判断することができる。第3にプログラム構造におけるボトルネックを特定することができる。第4に要求性能を満足するために、同時に実行するプログラムの構成やタスク数といったシステムに対する負荷をどのように設定すべきかを適切に判断することができ

る。第5にシステムにおけるスループット、レスポンスタイム、リソースの使用率といった細かい性能指標の予測が可能となる。

【0020】

【発明の実施の形態】まず、本発明の概要について説明する。本発明に係るマルチタスクシステムの性能予測システムは、プラットフォームとなる並列計算機の並列度、プロセッサの処理速度などのスペックを表すパラメータ、同時に処理されるプログラムの種類やそれらの数、各プログラムにおいてスレッド（thread）化がどのようになされているかを示すパラメータ、各プログラムにおいてセマフォ、クリティカルセクションがどのように採用されているかを示すパラメータなどを入力パラメータとし、これらのパラメータを基に、マルチプロセッサを（プロセッサの数の）複数個のサーバ（server）として捉え、それらのサーバがプロセッサシェアリング（processor sharing）でタスク、又はスレッドを処理するとして、セマフォやクリティカルセクションをトークン（token）を用いて、システムに対する負荷を客としてモデル化するモデル生成部を持つことを特徴としている。

【0021】このモデル生成部により生成されるモデルには、待ち行列理論の適用が可能であるため、陽に処理性能の向上度や性能指標を予測する式が求められ、この式から予測値が容易に求められるという効果が得られる。さらに、解析手段として待ち行列理論を用いることにより、必要な入力パラメータを大幅に削減できることを特徴としている。入力パラメータとして、システムの動作などの細かいデータが不必要となり、予測のための準備が容易になるという効果が得られる。

【0022】又、解析手段としては、待ち行列理論の代わりにシミュレーションを用いてもよく、この場合は数値的に処理性能の向上度や性能指標を予測することができる。又、モデル化においては、プラットフォームのリソースをさらに細かくモデル化してもよい。この場合は待ち行列理論を用いても陽な式としては処理性能の向上度や性能指標を予測することは一般にはできないが、近似計算法やシミュレーションを用いることにより精度よく数値的に予測することが可能である。

【0023】以下、本発明の実施の形態について添付図面を参照しながら説明する。図1は本発明に係るマルチタスクシステムの性能予測システムの第1の実施の形態の構成図である。図1を参照して、性能予測システムは、プラットフォームとなる並列計算機の並列度やプロセッサの処理速度などのスペック、及び同時に処理されるプログラムの種類やそれらの数、各プログラムにおいてスレッド化がどのようになされているかやセマフォ、クリティカルセクションがどのように採用されているかを示すデータが入力されるシステムデータ入力部11と、システムデータ入力部11からの情報を入力としモデル

を生成するモデル生成部12と、生成されたモデルから向上度やより細かい性能指標であるスループット、レスポンス、リソース使用率などの性能指標予測値を算出する予測実行部13と、予測実行部13から出力された値を可視的に表現する予測結果出力部14とを含んで構成されている。

【0024】図2は性能予測システムの動作の手順を示すフローチャートである。次に、この図2を参照しながら性能予測システムの動作について説明する。まず、プラットフォームとなる並列計算機のスペックがシステムデータ入力部11へ入力される(S1)。ここでのスペックとは並列計算機の並列度などをいう。次に、システムデータ入力部11へプログラムに関するデータが入力される(S2)。そのプログラムに関するデータとは、(1)同時に実行されるプログラムの種類およびそれらの数を示すデータ、(2)各プログラムがどのようなタスク、スレッドで構成されているかを示すデータ、(3)各プログラムがセマフォやクリティカルセクションをどのように用いているかを示すデータ、(4)各タスク、スレッドのプラットフォーム上のリソースの平均使用頻度や平均使用時間を示すデータである。これらの入力されたデータを基にモデル生成部12によりモデルが生成される(S3)。

【0025】図1におけるモデル生成部12では、

(1)プログラムのセマフォやクリティカルセクションなどの構造がトークンを用いてモデル化され、(2)実行状態にあるプログラムおよびスレッドが(待ち行列モデルでいうところの)客としてモデル化され、(3)並列計算機が、プロセッサシェアリングによりサービスを行う、並列度と等しい数の(待ち行列モデルでいうところの)サーバとしてモデル化され、これにより待ち行列網モデルが生成される。

【0026】この待ち行列網モデルが予測実行部13により解析される(S4)。このモデルの状態の変化は各タスク又はスレッドの処理速度が分かれば追うことができるということが待ち行列理論において知られている。この処理速度は、上記のようなシンプルなモデル化したことにより、算出することが可能である。予測実行部13では、これらの処理速度を算出した後、これらを用いてモデルの状態の定常分布、モデルのスループット、レスポンスタイムなどの性能予測値を待ち行列理論を用いて算出する。そして、この算出値を予測結果出力部14が受け取り、数値のみならず指定されればグラフなども用いてユーザに理解しやすい形式にして掲示する(S5)。

【0027】例えば次のようなシステムを例として取り上げてみる。図3はプログラムの実行タイミングの一例*

*を示すタイミングチャートである。同時に実行されるプログラムは1種類でその数は常時K(Kは2以上の整数)個であるとする。又、図3に示されるように各プログラムは、タスク1とタスク2とから構成され、この2つのタスク1、2の実行を繰り返すものとする。又、タスクi(iは1又は2とする)はh(i)個のセマフォで制御されている。つまり、同時にタスクiを実行できるプログラムはシステム全体でh(i)個である。又、並列計算機の並列度に関わらず、タスク1の処理は平均 $\mu(1)$ 秒/回だけプロセッサを使用し、タスク2の処理は $\mu(2)$ 秒/回だけプロセッサを使用するとする。又、並列度はnとする。

【0028】このようなシステムは、図4の待ち行列網モデルの模式説明図に示されるように、(1)K人の客21が網内を循環し、(2)客21は種類i(iは1又は2)のトークンを確保したのちにタスクiの処理をサービスステーション22で受け、処理を終了するとトークンを解放し、(3)トークンの割り当てはFIFO(先着先出し)の規律で行われ、(4)客21はタスク1とタスク2を交互に行い、(5)種類iのトークンはh(i)個あり、(6)サービスステーション22はプロセッサシェアリングで客21をサービスするn(nは正の整数)個のサーバ23で構成される、という待ち行列網モデルとしてモデル生成部12においてモデル化される。

【0029】さらに具体的に説明すると、「K人の客21」とはプログラム1~Kの各々をいい、客21は種類1のトークンを獲得した後に行う処理1と、種類2のトークンを獲得した後に行う処理2の2種類の処理を交互に繰り返す。ただし、各種類のトークンの数には制限があるため、トークンを獲得できるまで客21はバッファ24に並んで空きができるまで待つ。そして、トークンを獲得した客21はサービスステーション22に進み、n個のサーバ23からプロセッサシェアリングの規律でサービスを受ける。そして、サービスステーション22にて処理1を終了した客21は種類2のトークンを獲得するために移動し、処理2を終了した客21は種類1のトークンを獲得するために移動する。

【0030】この待ち行列網モデルを予測実行部13において解析する。タスクiを実行中の客数をX(i)で表現するならば、モデルの状態は

$$X = (X(1), X(2), \dots, X(K))$$

により表現することができる。この状態Xの変化の速度は(この状態Xに依存する)タスクiの処理速度として求めることができる。L(i) = min(h(i), X(i))と表すと、この例の場合、状態Xの時の各タスクの処理速度は、

$$\text{処理1の処理速度 } \tau(1 | X) = n * \mu(1) L(1) / \max(n, L(1) + L(2)) \quad \dots (2)$$

$$\text{処理2の処理速度 } \tau(2 | X) = n * \mu(2) L(2) / \max(n, L(1) + L(2))$$

1) + L(2)) ... (3)

と求めることができる。モデルの状態変化の速度を算出できれば状態Xとなる定常分布、モデルのスループット、レスポンスタイムを算出することは待ち行列理論を用いば容易なことである。例えばこの例におけるスル*

*ーブットλは、
【0031】
【数1】

$$\lambda = \frac{Ch(2)n\mu(2)}{\max(n, h(2))} \sum_{k=1}^K \prod_{i=1}^{k-1} \frac{\tau(2|i, K-i))}{\tau(1|i, K-i))}$$

ここで、

$$C = 1 / \sum_{k=0}^K \Pr(X(1) = k, X(2) = K - k),$$

$$\Pr(X(1) = k, X(2) = K - k) = \begin{cases} C & (k = 0) \\ C \prod_{i=1}^k \frac{\tau(1|i, K-i))}{\tau(2|i, K-i))} & (k \geq 1) \end{cases}$$

【0032】のように陽に算出することができる。速度向上率、効率、レスポンス、リソースの使用率などの他の性能指標も待ち行列理論を適用することで陽に予測することができる。予測実行部13はモデルの性能指標の計算結果を予測値として出力する。そして、この出力を予測結果出力部14が受け取り、数値をグラフなども用いてユーザに理解しやすい形式にして提示する。

【0033】次に第1の実施の形態の効果について説明する。本発明は、マルチタスク環境又はマルチスレッド環境における実際のシステムを適切にかつシンプルに理論解析が可能な待ち行列網モデルとしてモデル化しているために、マルチタスク環境又はマルチスレッド環境における並列計算機システムの性能指標を容易にかつ精度よく予測できる。又、適切なモデル化により、従来の手法に比べ様々な指標を予測することができる。

【0034】なお、上記実施の形態では、先に触れたように、予測実行部13における解析手段として、待ち行列理論のかわりにシミュレーションを用いてもよい。又、モデル化においては、プラットフォームのリソースをさらに細かく待ち行列でいうサーバとしてモデル化してもよい。この場合は待ち行列理論を用いても陽な式としては処理性能の向上度や性能予測指標を求めることはできないが、近似計算法やシミュレーションを用いることにより同様に精度よく数値的に算出することが可能である。

【0035】プラットフォームのリソースをさらに細かくモデル化した例を挙げることにする。図5はプラットフォームのリソースをさらに細かくした待ち行列網モデルの模式説明図である。なお、図4の模式説明図と同様の構成部分には同一番号を付し、その説明を省略する。予測の対象となるシステムは上記と同様であるが、プラットフォームとなる並列計算機のリソースとしてハードディスク32、34が夫々存在し、これらのリソースのタスク、スレッドによる競合も考慮に入れて性能指標の予測

を行う。

【0036】タスク1はハードディスク32を利用し、タスク2はハードディスク34を利用するとする。この場合は、図5に示されるように、図4で示された待ち行列網モデルにハードディスク32に相当するサービスステーション31とハードディスク34に相当するサービスステーション33を加えた形で待ち行列網モデルとしてモデル化すればよい。ここでは、サービスステーション31とサービスステーション33はともに1つのサーバと1つのバッファで構成されており、サービス規律をFIFOとしているが、リソースのモデル化はその特性を反映した形で適切に行う必要がある。又、予測のための解析手法は上記と同様である。

【0037】次に、第2の実施の形態について説明する。その基本的構成は第1の実施の形態と同様であるが、システムにおいて適切なプラットフォームのスペックを特定するために、プラットフォームのスペック（具体的には並列度など）を予測を行いたい範囲のパラメータとしてシステムデータ入力部11に与える。モデル生成部12は、プラットフォームのスペックをパラメータにした形でモデル化を行う。予測実行部13は、与えられた範囲でパラメータを変化させ、第1の実施の形態と同じ方法で予測値を算出する。予測結果出力部14はその計算結果を受取り、パラメータに対する予測値をユーザに理解しやすい形式にし表示を行う。ユーザはこの結果から、どのパラメータが適切であるかを判断する。つまり、適切なプラットフォームとなる並列計算機のスペックを判断する。

【0038】例えば、先の第1の実施の形態で用いたシステムに対して並列度nを1から16の範囲とした場合のスループットを予測し、適切な並列度を算出することを考える。ユーザは第1の形態における実施例と同様の手順でモデルを生成するが、並列度は1からnのパラメータであると指定した入力システムデータ入力部11

に対して行う。モデル生成部12は、並列度を1からnまでのパラメータとしてモデルを生成し、それらのモデルを予測実行部13においてパラメータの値を変えながら性能指標の予測値を算出する。そしてユーザに理解しやすい形にし、表示を行うが、この場合、並列度nに対するスループットを数値だけではなく、図6に示すようなグラフにして表示するとよい。図6を参照して、要求性能が秒当たり4件だとするならば、並列度は5以上でなければならないことがわかる。

【0039】次に第2の実施の形態の効果について説明する。本発明はプラットフォームのスペックをパラメータとし、そのパラメータに対する予測値を見ることができる。このため、システムが要求されている性能仕様を満たすために必要なプラットフォームのスペックを決定するために必要な客観的なデータ（性能指標）を予測値として得ることができる。

【0040】次に、第3の実施の形態について説明する。その基本的構成は第1の実施の形態と同様であるが、プログラムにおけるセマフォやクリティカルセクションの数や構成、各タスク、スレッドのプラットフォーム上のリソースの平均使用時間、平均利用頻度などといったプログラム構造に関するデータを予測を行いたい範囲のパラメータとしてシステムデータ入力部11に与える。

【0041】モデル生成部12は、プログラムデータをパラメータにした形でモデル化を行う。予測実行部13は、与えられた範囲でパラメータを変化させ、第1の実施の形態と同じ方法で予測値を算出する。予測結果出力部14はその計算結果を受け取り、パラメータに対する予測値をユーザに理解しやすい形式にし表示を行う。ユーザはこの結果から、どのパラメータが適切であるかを判断する。つまり、適切なプログラム構造を判断する。

【0042】例えば、先の第1の実施の形態で用いたシステムに対してタスク1のセマフォの数を1($h(1)=1$)とし、タスク2のセマフォの数 $h(2)$ をパラメータとし、1～ h まで変化させた場合のスループットを予測し、適切セマフォの数を算出することを考える。ユーザは第1の形態における実施例と同様の手順でモデルを生成するが、セマフォの数 $h(2)$ は1から h のパラメータであると指定した入力をシステムデータ入力部11に対して行う。モデル生成部12は、セマフォの数 $h(2)$ を1から h までのパラメータとしてモデルを生成し、それらのモデルを予測実行部13がパラメータの値を変えながら性能予測値を算出する。そして予測結果出力部14においてユーザに理解しやすい形にし表示を行う。この場合も $h(2)$ に対するスループットを数値だけではなく、グラフデータなども表示するとよい。

【0043】次に第3の実施の形態の効果を示す。本発明はプログラムにおけるセマフォの数などプログラム構造に関するデータをパラメータとし、そのパラメータに

対する予測値を見ることができる。このため、システムが要求されている性能仕様を満たすためにはプログラム構造をどのようにすればよいか判断するために必要な客観的なデータを得ることができる。

【0044】次に、第4の実施の形態について説明する。その基本的構成は第1の実施の形態と同様であるが、同時に実行されるプログラム又はタスクの数、スレッドの数などのシステムの負荷に関するデータを予測を行いたい範囲のパラメータとしてシステムデータ入力部11に与える。モデル生成部12は、このようなシステムの負荷をパラメータにした形でモデル化を行う。予測実行部13は、与えられた範囲でパラメータを変化させ、第1の実施の形態と同じ方法で予測値を算出する。予測結果出力部14はその計算結果を受け取り、パラメータに対する予測値をユーザに理解しやすい形式にし表示を行う。ユーザはこの結果から、どのパラメータが適切であるかを判断する。つまり、適切な負荷を判断する。

【0045】例えば、先の第1の実施の形態で用いたシステムに対してプログラムの数を1～ K まで変化させた場合のスループットを予測し、適切な同時に実行するプログラムの数を算出することを考える。ユーザは第1の形態における実施例と同様の手順でモデルを生成するが、プログラムの数を1から K のパラメータであると指定した入力をシステムデータ入力部11に対して行う。モデル生成部12は、プログラムの数 k を1から K までのパラメータとしてモデルを生成し、それらのモデルを予測実行部13がパラメータの値を変えながら性能予測値を算出する。そして予測結果出力部14はその計算結果を受け取り、ユーザに理解しやすい形にし表示を行う。

【0046】次に、第4の実施の形態の効果を示す。本発明はプログラムにおけるプログラムの数などシステムにおける負荷をパラメータとし、そのパラメータに対する予測値を見ることができる。このため、システムが要求されている性能仕様を満たすためにはシステムにおける負荷をどのようにすればよいか判断するために必要な客観的なデータを得ることができる。

【0047】次に、第5の実施の形態について説明する。第5の実施の形態は性能予測方法プログラムを記録した記録媒体に関するものである。図7は記録媒体及び記録媒体駆動装置の構成図である。図7を参照して、記録媒体駆動装置はCPU（中央処理装置）41と、入力部42と、記憶部43と、性能予測システム44とを含んで構成され、この駆動装置で記録媒体45を駆動する。

【0048】性能予測システム44は図1に示されるシステムデータ入力部11、モデル生成部12、予測実行部13及び予測結果出力部14から構成されている。又、記録媒体45には図2のフローチャートで示される

性能予測方法プログラムが予め記憶されている。

【0049】次に、この駆動装置の動作について説明する。まず、入力部42を介してCPU41に性能予測方法プログラムのロード（LOAD）命令が入力されると、CPU41は記録媒体45から性能予測方法プログラムを読み込み、その読込んだプログラムを記憶部43に書き込む。次に、入力部42を介してCPU41に性能予測方法プログラムのラン（RUN）命令が入力されると、CPU41は記憶部43から性能予測方法プログラムを読み込み、その読込んだプログラムにより性能予測システム44を制御する。その制御の内容については前述したので説明を省略する。

【0050】

【発明の効果】本発明による第1の発明によれば、並列計算機をプラットフォームとするマルチタスクシステムの性能予測システムであって、そのシステムはマルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化するモデル化手段と、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する性能指標予測手段とを含むため、まず第1にマルチタスク環境またはマルチスレッド環境における並列計算機の並列度に対する性能の向上度や性能指標を正しく予測することができる。次に第2にかけられるコストにふさわしい並列度、より具体的にはプロセッサの数の決定を適切に判断することができる。第3にプログラム構造におけるボトルネックを特定することができる。第4に要求性能を満足するために、同時に実行するプログラムの構成やタスク数といったシステムに対する負荷をどのように設定すべきかを適切に判断することができる。第5にシステムにおけるスループット、レスポンスタイム、リソースの使用率といった細かい性能指標の予測が可能となる。

【0051】又、本発明による第2の発明によれば、並列計算機をプラットフォームとするマルチタスクシステムの性能予測方法であって、その方法はマルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル*

*化する第1ステップと、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する第2ステップとを含むため第1の発明と同様の効果を奏する。

【0052】又、本発明による第3の発明によれば、並列計算機をプラットフォームとするマルチタスクシステムの性能予測方法プログラムを記録した記録媒体であって、その記録媒体にはマルチタスク環境でのリソースの競合を待ち行列網モデルを用いてモデル化する第1ステップと、そのモデル化されたリソースの競合を待ち行列理論を用いて解析し前記マルチタスクシステムの性能指標を予測する第2ステップとを含むプログラムが記録されているため第1の発明と同様の効果を奏する。

【図面の簡単な説明】

【図1】本発明に係るマルチタスクシステムの性能予測システムの第1の実施の形態の構成図である。

【図2】性能予測システムの動作の手順を示すフローチャートである。

【図3】プログラムの実行タイミングの一例を示すタイミングチャートである。

【図4】待ち行列網モデルの模式説明図である。

【図5】プラットフォームのリソースをさらに細かくした待ち行列網モデルの模式説明図である。

【図6】スループット対並列度特性図である。

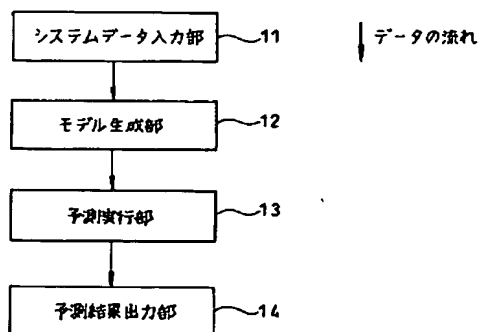
【図7】記録媒体及び記録媒体駆動装置の構成図である。

【図8】並列計算機をプラットフォームとするマルチタスク環境でのコンピュータシステムの一例を示す構成図である。

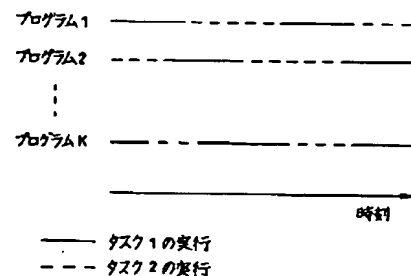
【符号の説明】

- 11 システムデータ入力部
- 12 モデル生成部
- 13 予測実行部
- 14 予測結果出力部
- 45 記録媒体

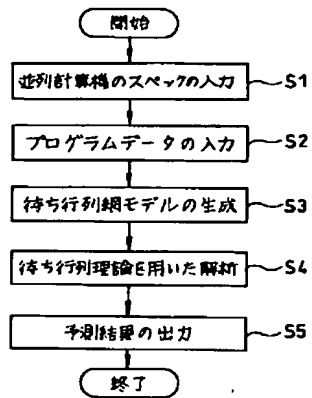
【図1】



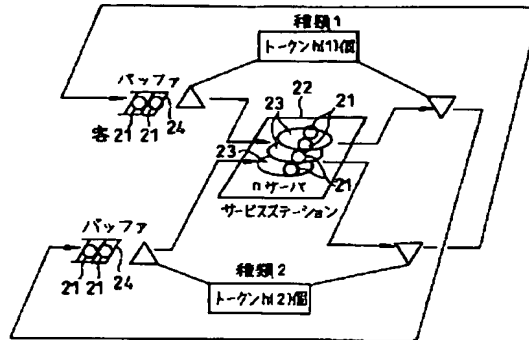
【図3】



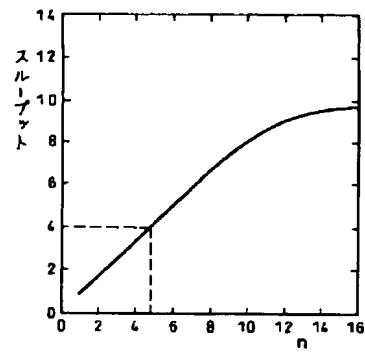
【図2】



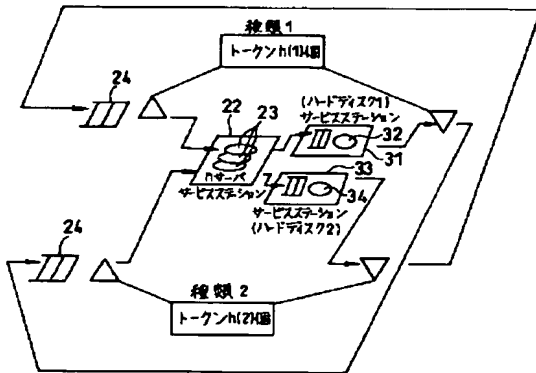
【図4】



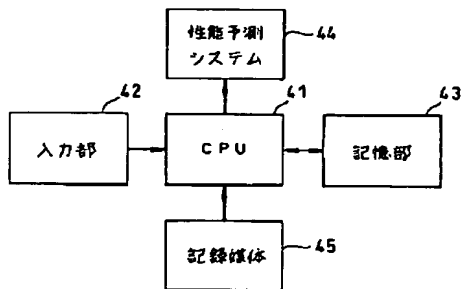
【図6】



【図5】



【図7】



【図8】

